

# ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT



УДК 681.3.681.5

DOI 10.12737/20225

## Применение генетического алгоритма для решения задачи покрытия множеств\*

**И. С. Коновалов<sup>1</sup>, В. А. Фатхи<sup>2</sup>, В. Г. Кобак<sup>3\*\*</sup>**<sup>1, 2, 3</sup> Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

## Application of genetic algorithm for the set-covering problem solution\*\*\*

**I. S. Kononov<sup>1</sup>, V. A. Fatkhi<sup>2</sup>, V. G. Kobak<sup>3\*\*</sup>**<sup>1, 2, 3</sup> Don State Technical University, Rostov-on-Don, Russian Federation

Рассматриваются взвешенная и невзвешенная задачи нахождения минимального покрытия множеств, а также ее применимость для решения важнейших оптимизационных практических задач, таких как размещение пунктов обслуживания, назначение экипажей на транспорте, проектирование интегральных схем и конвейерных линий. Цель статьи — описание методов повышения эффективности решения данной задачи. Сформулирован принцип работы генетического алгоритма и возможность использования его модификации в качестве метода решения задачи покрытия множеств. Рассматривается жадная стратегия Хватала для решения задачи покрытия. Для решения задач небольшого размера разработан алгоритм полного перебора в качестве точного алгоритма. Описан модифицированный генетический алгоритм, разработанный Нгуеном М. Х. Создано программное средство для сравнения производительности этих алгоритмов. Сделаны выводы о том, что решение задачи покрытия множеств разработанной модификацией генетического алгоритма более эффективно, чем генетическим алгоритмом Нгуена М. Х. и жадной стратегией, причем в задачах небольшого размера полученные результаты отличаются небольшой погрешностью.

The weighed and unweighed minimal set-cover problem, its applicability for the solution of the major optimization practical tasks, such as arrangement of service points, assignment of crews in transport, as well as the integrated-circuit and conveyer lines designing is considered. The paper objective is to describe methods of improving efficiency of this task solution. The principle of operation of the genetic algorithm and the applicability of its modification as a method of the solution to the set-cover problem are defined. Greedy strategy of Chvatal for the set-cover problem solution is considered. An exhaustive algorithm as an exact algorithm for the solution of small-size tasks is developed. The modified genetic algorithm developed by Nguyen M. H. is described. A software tool for comparing the performance of these algorithms is created. It is concluded that the solution of the set-cover problem by our genetic algorithm modification is more effective than by the genetic algorithm of Nguyen M. H. or the greedy strategy. And the results obtained in small-size tasks are noted for insignificant error.

**Ключевые слова:** генетический алгоритм, задача покрытия множеств, модель Голдберга, алгоритм Нгуена М. Х., жадная стратегия Хватала, полный перебор.

**Keywords:** genetic algorithm, set-covering problem, Goldberg's model, algorithm of Nguyen M.H., greedy strategy of Chvatal, exhaustive enumeration.

**Введение.** Задача о покрытии множества является классическим вопросом информатики и теории сложности. Данная задача обобщает  $NP$ -полную задачу о вершинном покрытии (и потому также является  $NP$ -сложной).

К задаче о покрытии можно свести многие задачи дискретной оптимизации: стандартизации, упаковки и разбиения множества, построения расписаний. Известна также и обратная сводимость задачи о покрытии к этим задачам.

На практике задачи о покрытии возникают при размещении пунктов обслуживания, в системах информационного поиска, при назначении экипажей на транспорте, проектировании интегральных схем и конвейерных линий и т. д.

**Постановка задачи.** Дано множество  $U$  из  $n$  элементов, и набор подмножеств  $U$ ,  $S = \{S_1, \dots, S_k\}$ . Каждому подмножеству  $S_i$  сопоставлена некоторая неотрицательная стоимость  $c$ :  $S \rightarrow Q^+$ .  $S' \subseteq S$  является покрытием, если любой элемент из  $U$  принадлежит хотя бы одному элементу из  $S'$  [1].

Задача о покрытии множествами заключается в нахождении набора подмножеств, покрывающего все множество  $U$  и имеющего минимально возможный вес (в случае взвешенной задачи) или минимально возможное число подмножеств (в случае невзвешенной задачи).

Для упрощения понимания задачи рассмотрим следующий пример. Необходимо собрать команду специалистов для корабля. Члены команды должны обладать в совокупности всеми требуемыми навыками, но количество членов команды должно быть минимальным. Это невзвешенная задача покрытия, т. е. «вес» каждого члена группы оди-

\* Работа выполнена в рамках инициативной НИР.

\*\* E-mail: xigorx92@mail.ru, fatkhi@mail.ru, valera33305@mail.ru

\*\*\* The research is done within the frame of the independent R&D.

наков и поэтому не важен. Если же каждому члену команды поставить в соответствие какую-то величину (вес), например, опыт работы, то задача станет взвешенной.

Можно представить задачу в матричном виде [2]. Пусть  $A=(a_{ij})$  — произвольная матрица размера  $m \times n$  с элементами  $a_{ij} \in \{0,1\}$  без нулевых строк и столбцов. Будем говорить, что в  $A$  строка  $i$  покрывается столбцом  $j$ , если  $a_{ij}=1$ . Подмножество столбцов называется покрытием, если в совокупности они покрывают все строки матрицы  $A$ . Пусть каждому столбцу поставлено в соответствие положительное число  $c_j$ , называемое весом столбца. Требуется найти покрытие минимального суммарного веса. Вводя переменные  $x_j$ , равные 1, если столбец  $j$  входит в искомое покрытие, и равные 0 в противном случае, приходим к следующей формулировке задачи о покрытии:

$$\sum_{j=1}^n c_j x_j \rightarrow \min,$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \geq 1, i=1, \dots, m, x_j \in \{0,1\}, j=1, \dots, n.$$

**Обзор алгоритмов решения.** Для решения задачи покрытия множеств разработано множество алгоритмов, которые можно разделить на следующие классы: приближенные алгоритмы с априорной оценкой, эвристические алгоритмы, точные алгоритмы [3].

Одним из первых приближенных алгоритмов является жадный алгоритм. Для задачи о покрытии с произвольными весами В. Хватал предложил модификацию жадного алгоритма. Поскольку эвристики носят вероятностный характер, их сложность невозможно оценить априорно. К ним можно отнести методы лагранжевой релаксации, генетические алгоритмы, поиск с запретами, алгоритмы муравьиной колонии, нейронные сети. Примером точных алгоритмов служит метод ветвей и границ.

В данной статье авторы рассматривают жадный алгоритм Хватала, модифицированную модель Голдберга генетического алгоритма и эвристику, разработанную Нгуеном М. Х. Кроме того, показано, насколько результаты работы модифицированной модели Голдберга близки к точному решению, полученному с помощью алгоритма точного перебора.

**Жадный алгоритм.** В 1979 году В. Хватал предложил жадный алгоритм для задачи о покрытии множествами [4, 5].

Алгоритм Хватала ( $U, S, c: S \rightarrow Q^+$ ):

1)  $C \leftarrow \emptyset, Sol \leftarrow \emptyset$ ;

2) While  $C \neq U$  do:

Найти  $S_i \in S - Sol$ , у которого

$\alpha_i = c(S_i) / |S_i - C|$  минимально.

$Sol \leftarrow Sol \cup \{S_i\}$ .

$C \leftarrow C \cup S_i$  ( $S_i$  – самое эффективное),

$price(e) = \alpha_i$  для всех  $e \in S_i - C$ .

3) Output ( $Sol$ ).

Рассмотрим принцип работы этого алгоритма. На каждой итерации выбираем самое эффективное множество, удаляем покрытые элементы и продолжаем до тех пор, пока не будут покрыты все элементы. Пусть  $C$  — это множество элементов, уже покрытых на предыдущих итерациях. Для каждого множества  $S_i$  определим его эффективность как  $\alpha_i = c(S_i) / |S_i - C|$ . Эффективность множества равна средней стоимости, с которой покрываются элементы этого множества, еще не покрытые на предыдущих итерациях.

**Генетический алгоритм. Общая схема работы.** Предложенные в 1975 году Джоном Холландом генетические алгоритмы (ГА) основаны на принципах естественного отбора и наследования и относятся к стохастическим методам [6–9]. Эти алгоритмы успешно применяются в различных областях деятельности (экономика, физика, технические науки и т. п.), их используют для решения многих оптимизационных задач. На рис. 1 представлена общая схема работы ГА.

**Применение генетического алгоритма для решения задачи покрытия множеств.** Представленная схема является общим алгоритмом для решения многих задач, и при применении её к конкретной задаче необходимо выбрать механизм кодирования параметров в гены особи, оптимизационную функцию, условие останова [1]. Авторы модифицировали модель Голдберга и применили ее для решения задачи покрытия множеств. В данном случае оптимизационной функцией будет являться минимизация веса покрытия, а условием останова будет неизменность лучшего решения в течение заданного числа поколений.

Рассмотрим механизм кодирования особи. Каждый индивид  $k$  представлен хромосомой, являющейся  $n$ -мерным вектором  $x^k$ , у которого  $j$ -й элемент  $x_j^k$  принимает значение 1, если подмножество  $S_j$  входит в покрытие, и принимает значение 0, если иначе.

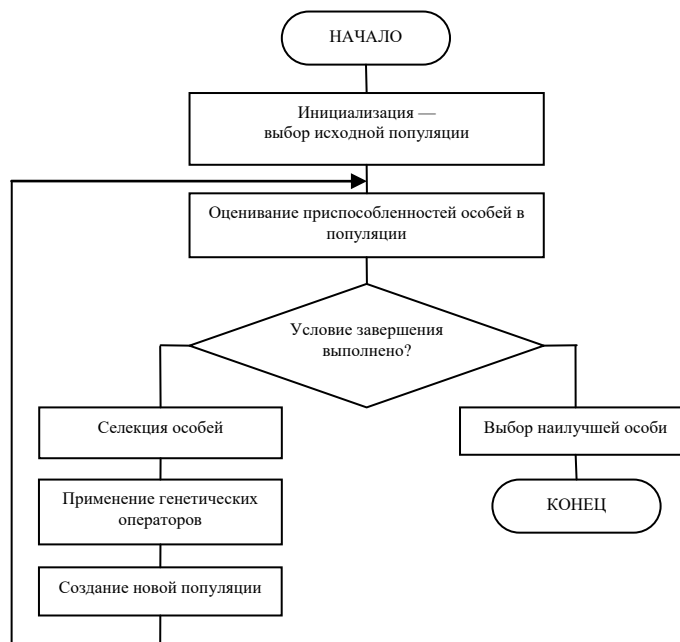


Рис. 1. Общая схема ГА

С таким представлением степень приспособленности  $f_k$  индивида  $x^k$  может быть рассчитана следующим образом:

$$f_k = \sum_{j=1}^n c_j x_j^k,$$

где  $c_j$  — стоимость подмножества  $S_j$ .

Таким образом, оптимизационная функция выглядит как  $f_k \rightarrow \min$ .

Для выбора родительских особей используется случайный отбор. В алгоритме используется тип мутации, основанный на изменении случайного гена на противоположное значение. Оператор скрещивания точечный. Выбираются пары хромосом из родительской популяции. Далее для каждой пары отобранных таким образом родителей разыгрывается позиция гена (локус) в хромосоме, определяющая так называемую точку скрещивания —  $l_k$ . В результате скрещивания пары родительских хромосом получается следующая пара потомков:  $P1$  — потомок, хромосома которого на позициях от 1 до  $l_k$  состоит из генов первого родителя, а на позициях от  $l_k + 1$  до  $L$  — из генов второго родителя;  $P2$  — потомок, хромосома которого на позициях от 1 до  $l_k$  состоит из генов второго родителя, а на позициях от  $l_k + 1$  до  $L$  — из генов первого родителя.

Начальное поколение состоит из особей, сформированных случайно с помощью алгоритма, подобному жадному алгоритму.

Очень важная деталь работы генетического алгоритма в том, что при скрещивании и мутации могут появиться особи, соответствующие покрытия которых не существуют, то есть получают недопустимые решения. Алгоритм проверяет, существует ли покрытие, и если нет, то пытается, в случае скрещивания, выбрать другую вторую родительскую особь, а в случае мутации — выбрать другой ген для его инвертирования. Если и это не «исправит» особь, родитель выбирается заново случайным образом. Потомок заменяет случайно выбранную особь, если его приспособленность выше.

Рассмотрим на примере, как работает генетический алгоритм и как он избавляется от недопустимых решений. Задача представлена в виде матрицы размером  $10 \times 10$ , заполненной "0" и "1". Столбцы матрицы — это подмножества множества  $U$ , а строки — элемента множества  $U$ . Таким образом, множество  $U = \{x_0, \dots, x_9\}$ , и оно состоит из 10 подмножеств  $S = \{S_0, \dots, S_9\}$ . "1" в матрице обозначает, что соответствующее столбцу подмножество покрывает соответствующий строке элемент. Не будем учитывать веса подмножеств, т. е. задача невзвешенная.

$x_0$	0	1	0	0	1	0	1	1	1	1
$x_1$	1	0	1	0	1	1	0	0	1	0
$x_2$	0	1	0	1	0	0	1	1	0	1
$x_3$	1	0	1	0	1	1	0	1	0	0
$x_4$	1	1	0	1	0	1	1	1	0	0
$x_5$	0	0	1	1	0	0	0	0	0	1
$x_6$	1	0	0	0	1	0	1	0	1	0
$x_7$	0	0	1	0	1	0	0	0	0	1
$x_8$	0	0	1	1	1	1	0	1	0	1
$x_9$	0	1	1	0	1	0	1	1	1	1
	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$

Рис. 2 Матрица подмножеств исходного множества

Используем для поиска покрытия наименьшей мощности ГА с 10 особями.

Каждая особь соответствует определенному покрытию, поэтому длина особи равна 10 и каждый ген  $Ch=\{Ch_0,...,Ch_9\}$  соответствует определенному подмножеству. Если ген особи равен "1", то подмножество входит в покрытие, если "0", то не входит.

На начальном шаге алгоритма особи формируются случайным образом. Представим особи и их гены в виде массива, где строка — особь, столбец — ген особи.

особь 0	1	1	1	0	0	0	0	0	0	0
особь 1	1	0	0	0	1	0	0	0	0	1
особь 2	1	0	0	1	0	1	0	0	0	1
особь 3	1	1	1	0	0	0	0	1	0	0
особь 4	0	0	0	1	1	0	1	0	0	0
особь 5	1	0	0	1	1	0	0	0	0	0
особь 6	0	0	0	0	0	0	0	0	0	0
особь 7	1	0	1	0	0	1	0	1	0	1
особь 8	0	0	0	1	1	0	1	1	0	0
особь 9	1	1	1	0	0	1	0	0	1	0
	$Ch_0$	$Ch_1$	$Ch_2$	$Ch_3$	$Ch_4$	$Ch_5$	$Ch_6$	$Ch_7$	$Ch_8$	$Ch_9$

Рис. 3 Формирования начальной популяции

На следующем шаге случайно выбрана родительская особь "1". С некоторой вероятностью к ней применяется оператор одноточечного скрещивания, поэтому случайно выбирается вторая родительская особь "2". Точка скрещивания = ген №3. После скрещивания получаем потомка:

1	0	0	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

Покрытие, соответствующее этому потомку, существует. С некоторой вероятностью к потомку применяется оператор одноточечной мутации. Случайным образом выбран ген №9, он меняет свое значение на противоположное.

1	0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Но такого покрытия не существует, алгоритм пытается выбрать другой ген, выбран ген №2. Особь-потомок:

1	0	1	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

Такое покрытие существует. Далее в поколении случайно выбирается особь №9, которую, возможно, заменит потомок. Приспособленность потомка равная 4 выше приспособленности особи №9 равной 5, поэтому он ее заменяет.

Все описанные шаги выбора родительских особей и применения к ним операторов скрещивания и мутации повторяются, пока не будет найдено решение. Алгоритм заканчивает свою работу, если решение не может быть улучшено за заданное число поколений.

**Модификация генетического алгоритма Нгуена Минь Ханга.** В статье [10] Нгуен Минь Ханг показал свой подход к решению задачи покрытия множеств, основанный на генетическом алгоритме. Особенностью данного подхода является использование модифицированной для задачи покрытия классической модели ГА. Для выбора родительских пар используется метод пропорционального отбора. Каждой особи ставится в соответствие вероятность выбора этой особи, рассчитанная по ее эффективности. Разработан новый оператор кроссовера, суть которого в том, что соответствующий ген для особи-потомка определяется в зависимости от степени приспособленности родителя и от частоты появления этого гена в популяции. Кроме того, используется переменная частота мутации. При применении операторов кроссовера и мутации над индивидами допустимость решения может быть нарушена (т. е. полученный набор не является покрытием множества). Для устранения этого явления предлагается эвристический алгоритм восстановления допустимости решения, одновременно являющийся локальным оптимизационным шагом для увеличения общей эффективности генетического алгоритма.

**Анализ производительности алгоритмов.** Исследуем вопрос, какой алгоритм получает более точные решения, Голдберга или Нгуена М. Х. В целях сравнения эффективности этих алгоритмов по весам найденных покрытий и временным затратам было разработано программное средство на языке C#. Представлены результаты экспериментов на ПК следующей конфигурации: ОС *Microsoft Windows 10 Pro* x64, процессор *Intel(R) Core(TM) i5-2500K CPU 3.30GHz*, оперативная память 6 Гб. Было проведено по 100 экспериментов с различными матрицами размером  $n \times m$ , где  $n$  — количество подмножеств множества  $U$ ,  $m$  — мощность (количество элементов) множества  $U$ . Матрицы заполняются случайным образом "0" и "1".

- Коэффициент заполненности матрицы подмножеств единицами  $p=0.5$ .
- Интервал распределения весов подмножеств для взвешенной задачи  $1 \dots 200$  (целые числа), формируются случайным образом.
- Количество подмножеств и количество элементов множества  $U$  принимает значения  $30 \times 30$ ,  $50 \times 50$ ,  $100 \times 100$ ,  $150 \times 150$ ,  $200 \times 200$

Приведены результаты сравнения модели Голдберга с различным числом особей (50, 100, 150, 200), жадного алгоритма Хватала и эвристики Нгуена М. Х. (50, 100, 150, 200 особей). Параметры для ГА были заданы следующие:

- вероятность скрещивания = 1,
- вероятность мутации = 1,
- условие останова = 100 поколений,
- тип оператора скрещивания — одноточечное,
- тип мутации — одноточечная.

Результаты сравнения алгоритмов для взвешенной задачи по весам покрытий представлены в таблице 1, по времени работы — в таблице 2, для невзвешенной задачи — в таблицах 3 и 4 соответственно.

Таблица 1

Веса покрытий (взвешенная задача)

Алгоритм	30x30	50x50	100x100	150x150	200x200
ГА 50	88,57	65,67	48,98	40,97	32,72
ГА Нгуен50	91,66	67,44	53,67	51,92	50,84
ГА 100	88,51	63,12	46,3	39,57	31,99
ГА Нгуен100	90,34	66,13	49,36	43,58	42,5
ГА 150	87,22	62,76	45,93	38,95	31,25
ГА Нгуен150	89,78	65,71	49,44	41,97	40,21
ГА 200	87,05	62,83	46,22	38,35	30,74
ГА Нгуен200	88,94	65,57	49,54	41,47	38,81
Жадный	98,14	74,56	70,66	91,99	110,24

Таблица 2

Временные затраты работы алгоритмов в мс (взвешенная задача)

Алгоритм	30x30	50x50	100x100	150x150	200x200
ГА 50	254,2886	751,8143	3806,738	9814,355	19801,82
ГА Нгуен50	1850,67	1844,317	1881,946	1957,975	3506,081
ГА 100	488,2043	1387,882	7508,047	18293,5	35186,1
ГА Нгуен100	1931,191	2020,152	2308,606	4333,37	5555,818
ГА 150	703,5788	2036,386	10085,16	25527,69	49748,35
ГА Нгуен150	2065,456	2040,725	2381,178	4973,016	8947,126
ГА 200	926,0862	2684,479	12662,96	32821,01	65370,57
ГА Нгуен200	2075,047	2147,794	5029,717	8576,225	12329,05
Жадный	0,101141	0,216815	0,922629	2,186599	4,014783

Таблица 3

Веса покрытий (невзвешенная задача)

Алгоритм	30x30	50x50	100x100	150x150	200x200
ГА 50	2,96	3,23	4,17	4,84	5,1
ГА Нгуен50	3,28	3,94	4,93	5,45	5,89
ГА 100	2,97	3,12	4	4,71	4,98
ГА Нгуен100	3,05	3,78	4,76	5,26	5,69
ГА 150	2,96	3,03	3,93	4,59	4,98
ГА Нгуен150	3,01	3,73	4,61	5,1	5,62
ГА 200	2,96	3	3,95	4,47	4,92
ГА Нгуен200	3,01	3,7	4,54	5,04	5,61
Жадный	4,66	5,44	6,49	6,92	7,48

Таблица 4

Временные затраты работы алгоритмов в мс (невзвешенная задача)

Алгоритм	30x30	50x50	100x100	150x150	200x200
ГА 50	202,2554	565,297	2066,982	4261,727	7823,984
ГА Нгуен50	1612,93	1629,785	1603,908	1712,292	2627,43
ГА 100	401,6485	1181,412	3979,194	8521,661	14765,89
ГА Нгуен100	1608,363	1651,991	1668	3300,692	4343,407
ГА 150	582,8104	1625,455	5838,935	12848,03	20946,68
ГА Нгуен150	1612,511	1665,563	1679,632	3421,566	5561,238
ГА 200	770,9581	2138,706	7388,203	17932,41	28690,47
ГА Нгуен200	1604,523	1674,239	3351,328	4862,393	6814,878
Жадный	0,102653	0,2312	0,946716	2,236784	4,10912

**Алгоритм полного перебора (брутфорс).** Полный перебор (*brute force*) — метод решения математических задач. Относится к классу методов поиска решения исчерпыванием всевозможных вариантов. Сложность полного перебора зависит от количества всех возможных решений задачи.

Любая задача из класса *NP* может быть решена полным перебором. При этом, даже если вычисление целевой функции от каждого конкретного возможного решения задачи может быть осуществлено за полиномиальное время, в зависимости от количества всех возможных решений полный перебор может потребовать экспоненциального времени работы.



Суть алгоритма полного перебора для решения задачи покрытия заключается в переборе всех возможных сочетаний разной длины подмножеств исходного множества и, если очередное сочетание является покрытием и его вес минимален, необходимости сохранить его. Из теории множеств известно, что число всех подмножеств множества из  $n$  элементов равно  $2^n$ . Другими словами

$$\sum_{k=0}^n c_n^k = 2^n.$$

Итак, чтобы получить точное решение задачи размерности  $25 \times 25$ , пришлось перебрать  $2^{25} = 33554432$  вариантов, для этого потребовалось около 2–3 часов, но задача немного большей размерности  $30 \times 30$  включает  $2^{30} = 1073741824$  вариантов, что уже затруднительно для расчетов.

Для ускорения полного перебора программное средство было модифицировано для распараллеливания вычислений с помощью многопоточности.

Решения, полученные модификацией модели Голдберга с 200 особями, были сравнены с точными решениями, полученными брутфорсом, в задачах размерности до  $25 \times 25$ . Усредненные результаты по 100 экспериментам взвешенной задачи приведены в таблице 5.

Таблица 5

Весы покрытий точного решения и ГА

pxm	ГА	Точное решение
20x20	108,01	107,95
25x25	88,94	88,02

**Заключение.** Проведено сравнение производительности различных алгоритмов применительно к взвешенной и невзвешенной задачам покрытия множеств. Обе модификации генетического алгоритма намного превосходят жадный алгоритм по весовым показателям получившихся покрытий, но по параметру трудоемкости жадный алгоритм выигрывает с огромным отрывом. Если важна скорость вычислений, то в задачах небольшой размерности можно использовать жадный алгоритм. Чем больше особей в поколении использует ГА, тем лучше результат весов покрытий. ГА с 200 особями лучший из рассмотренных. Но, естественно, чем больше особей, тем дольше работает алгоритм. По сравнению с эвристикой Нгуена М. Х. модифицированная модель Голдберга показывает лучшие результаты, что более заметно в задачах большой размерности, но при этом она использует больше временных ресурсов, ведь операторы скрещивания и мутации с вероятностью 100% применяются для всех особей поколения поочередно, в отличие от механизма Нгуена М. Х. Кроме того, в задачах небольшой размерности ГА на основе модели Голдберга получает решения, очень близкие к точным.

#### Библиографический список

1. Коновалов, И. С. Сравнительный анализ работы жадного алгоритма Хватала и модифицированной модели Голдберга при решении взвешенной задачи нахождения минимального покрытия множеств / И. С. Коновалов, В. А. Фатхи, В. Г. Кобак // Труды СКФ МТУСИ. — 2015. — Ч. I. — С. 366–370.
2. Еремеев, А. В. Генетический алгоритм для задачи о покрытии / А. В. Еремеев // Дискретный анализ и исследование операций. — 2000. — Т. 7, № 1. — С. 47–60.
3. Еремеев, А. В. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования / А. В. Еремеев, Л. А. Заозерская, А. А. Колоколов // Дискретный анализ и исследование операций. — 2000. — Т. 7, № 2. — С. 22–46.
4. Кононов, А. В. Приближенные алгоритмы для  $NP$ -трудных задач / А. В. Кононов, П. А. Коконова. — Новосибирск : Новосиб. гос. ун-т., 2014. — 117 с.
5. Chvatal, V. A greedy heuristic for the set-covering problem // Mathematics of Oper. Res. — 1979. — V. 4, № 3. — P. 233–235.
6. Holland, J. H. Adaptation in Natural and Artificial Systems. The University of Michigan Press, 1975. — P. 245.
7. Goldberg, D. E. Genetic algorithms in search, optimization and machine learning. Reading, MA : Addison-Wesley, 1989. — P. 432.
8. Батищев, Д. И. Генетические алгоритмы решения экстремальных задач / Д. И. Батищев. — Н. Новгород : Нижегородский гос. ун-т., 1995. — 69 с.
9. Гладков, Л. А. Генетические алгоритмы / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. — Москва : Физматлит, 2010. — 368 с.
10. Нгуен, М. Х. Применение генетического алгоритма для задачи нахождения покрытия множества // Динамика неоднородных систем. — 2008. — Т. 33., Вып. 12. — С. 206–219.

## References

1. Kononov, I.S., Fatkhi, V.A., Kobak, V.G. Sravnitel'nyy analiz raboty zhadnogo algoritma Khvatala i modifitsirovannoy modeli Goldberga pri reshenii vzveshennoy zadachi nakhozheniya minimal'nogo pokrytiya mnozhestv. [Comparative analysis of work greedy algorithm of Chvatal and modified Goldberg models weighted in solving the problem of finding minimal coverings of sets.] Trudy SKF MTUSI, 2015, Part I, pp. 366–370 (in Russian).
2. Ereemeev, A.V. Geneticheskiy algoritm dlya zadachi o pokrytii. [A generic algorithm for the covering problem.] Discrete Analysis and Operations Research, 2000, vol. 7, ser. 2, no. 1, pp. 47–60 (in Russian).
3. Ereemeev, A.V., Zaozerskaya, L.A., Kolokolov, A.A. Zadacha o pokrytii mnozhestva: slozhnost', algoritmy, eksperimental'nye issledovaniya. [The set covering problem: complexity, algorithms, and experimental investigations.] Discrete Analysis and Operations Research, 2000, vol. 7, ser. 2, no. 2, pp. 22–46 (in Russian).
4. Kononov, A.V., Kononova, P.A. Priblizhennyye algoritmy dlya NP-trudnykh zadach. [Approximate algorithms for NP-hard problems.] Novosibirsk: Novosibirsk State University, 2014, 117 p. (in Russian).
5. Chvatal, V. A greedy heuristic for the set-covering problem. Mathematics of Oper. Res., 1979, vol. 4, no. 3, pp. 233–235.
6. Holland, J. H. Adaptation in Natural and Artificial Systems. The University of Michigan Press, 1975, 245 p.
7. Goldberg, D. E. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989, 432 p.
8. Batishchev, D.I. Geneticheskie algoritmy resheniya ekstremal'nykh zadach. [Genetic algorithms for solving extreme problems.] N. Novgorod: State University of Nizhni Novgorod, 199, 69 p. (in Russian).
9. Gladkov, L.A., Kureychik, V.V., Kureychik, V.M. Geneticheskie algoritmy. [Genetic algorithms.] Moscow: Fizmatlit, 2010, 368 p. (in Russian).
10. Nguyen, M.H. Primenenie geneticheskogo algoritma dlya zadachi nakhozheniya pokrytiya mnozhestva. [Application of genetic algorithm to the problem of finding cover sets.] Dinamika neodnorodnykh system, 2008, vol. 33, iss. 12, pp. 206–219 (in Russian).

Поступила в редакцию 08.02.2016

Сдана в редакцию 09.02.2016

Запланирована в номер 07.07.2016